ISSN 1561-5359. Штучний інтелект, 2017, № 2

УДК 681.3

А.М. Глибовець
Національний університет «Києво-Могилянська академія», Україна
вул. Сковороди, 2, м. Київ, 04070

АВТОМАТИЗОВАНИЙ ПОШУК ІМЕНОВАНИХ СУТНОСТЕЙ У НЕРОЗМІЧЕНИХ ТЕКСТАХ УКРАЇНСЬКОЮ МОВОЮ

A.M. Glybovets
National university «Kyiv-Mohyla academy», Ukraine
2, Scovorody str., Kyiv, 04070

AUTOMATED SEARCH OF NAMED ENTITIES IN UNMARKED UKRAINIAN TEXTS

У роботі описано створений та реалізований алгоритм пошуку іменованих сутностей у текстах українською мовою. Створені програмні інструменти дозволяють виділяти іменовані сутності та зв'язки між ними в графічному режимі. Утиліту реалізовано у вигляді веб-застосунку. За допомогою цього програмного інструментарію створено корпус анотованих NER сутностей текстів у кількості 122 тексти. Проставлено такі види сутностей як *персони*, *організації* та *географічні об'єкти*. Корпус складається з 2731 іменованої сутності.

Ключові слова: іменовані сутності, обробка природного тексту, виділення іменованих сутностей.

The paper describes the created and implemented algorithm of the search for named entities in the texts in the Ukrainian language. The software tools created on the basis of them allow to allocate the named entities and connections between them in graphic mode. The utility is implemented as a web application. With the help of this software tool, a body of annotated NERs of texts of 122 texts was created. There are such kinds of entities as persons, organizations and geographical objects. The body consists of 2,731 named entities.

Keywords: named entities, natural language processing, allocation of named entities.

Вступ

Обробка природної мови належить до найактуальніших і найскладніших завдань комп'ютерної лінгвістики. Алгоритми з автоматизації обробки текстів природною мовою досягли значного прогресу в останній час, про що свідчать результати таких конференцій як CoNLL[1] та MUC[2].

На момент проведення досліджень у відкритому доступі не було знайдено робіт у напрямку пошуку іменованих сутностей у текстах українською мовою.

Виділення іменованих сутностей

Задля семантичної цілісності тексту та його зв'язаності автори вдаються до використання різних типів повторів означень деякого об'єкту – кореферентів, які комплексно та різнобічно характеризують один і той же референт, тобто об'єкт, про який йде мова. Термін кореферентність (лат. co- – префікс, що означає сумісність; лат. referent – той що зіставляє) вживається для позначення предмета думки, з яким співвідноситься певне мовне вираження, відображене у свідомості елемента об'єктивної дійсності [3].

Наведемо приклад кореферентного зв'язку:

У субботу (1) Микола хотів піти до університетської бібліотеки (2), проте, вона (2) того дня (1) не працювала.

У наведеному реченні два кореферентних зв'язки: бібліотека – вона, та субота – той день. Ці дві групи слів відносяться, відповідно, до одного й того ж самого об’єкта і можуть бути взаємозамінені один одним.

Пошук кореферентних зв'язків є лише однією ланкою в NLP процесі та потребує виконання багатьох попередніх етапів. Розглянемо загальну схему задачі пошуку кореферентних зв’язків, зображену на Рис.1.

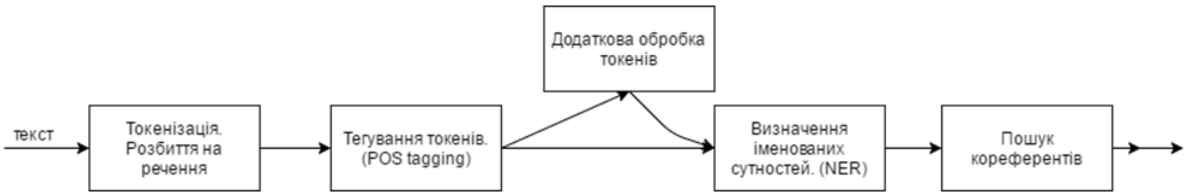


Рис.1. Пошук кореферентних зв'язків

На вхід програмного модуля подається текст, який обробляється токенізатором-застосунком, що розбиває текст на токени.

У рамках цієї роботи було розроблено токенізатор для української мови, який базується на PCRE [4] правилах. Список токенів, які він може обробити, наведено в таблиці 1.

Таблиця 1. Список правил для токенізатора української мови

<table>
<tr><th>Назва</th><th>Regex правило</th></tr>
<tr><td>CyrillicToken</td><td>[a-яєёїґі][a-яёєїґі\\'\\-’]*</td></tr>
<tr><td>EmailToken</td><td>[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\\.[a-zA-Z0-9-.]+</td></tr>
<tr><td>EndOfLineToken</td><td>[\\n\\r]+</td></tr>
<tr><td>FloatToken</td><td>[+-]?[\\d]+[\\.\\,][\\d]+</td></tr>
<tr><td>IntToken</td><td>[+-]?\\d+</td></tr>
<tr><td>LatinToken</td><td>[a-z][a-z\\'\\-]*</td></tr>
<tr><td>PhoneToken</td><td>(\\+)?([-\\s_()]?\\d[-\\s_()]?){10,14}</td></tr>
<tr><td>PunctuationToken</td><td>[\\—!\"\\#$%&'()*+,\\-./:;<=>?@\\[\\]^_`{|}~\\\\]"</td></tr>
<tr><td>QuoteToken</td><td>[\\\"\\'\\«\\»\\„\\“]</td></tr>
</table>

Була розроблена гнучка архітектура токенізатора мовою Java, що дозволяє з легкістю додавати нові типи токенів або ж модифікувати набір токенів під свої потреби.

На противагу великій кількості систем, де пунктуація тексту відкидається, ми залишаємо текст повністю в такому ж вигляді, в якому він був отриманий. Єдиним винятком з цього правила є те, що усі додаткові пробіли між словами видаляються, залишаючи тільки один.

Дуже важливо правильно розбити текст на речення, адже, від цього залежить значення ознак, які потім будуть використовуватися при побудові моделей

машинного навчання. Для вирішення цього завдання ми застосували готову бібліотеку JLanguageTool [5].

Наступним етапом є POS Tagging – тегування частин мови та іншої граматичної інформації в токенах. Ми скористались JLanguageTool – програмним за стосунком, який має Java API і має можливість тегувати слова за частинами мови та проставляти граматичні властивості. У роботі з JLanguageTool було помічено досить великий недолік – багато слів отримують значну кількість форм вживання, тобто алгоритм не може зняти семантичну неоднозначність. Це може призводити до погіршення роботи усіх подальших етапів. Проте, оскільки альтернативи для української мови немає – користуємось тим, що є.

Після протегування токенів POS-тегувальником, вони можуть бути додатково оброблені з метою привнесення до них якоїсь додаткової інформації. Наприклад, ми до токенів додававали тег «Person/Position» у випадку, якщо нормальна форма токена збігалася зі словом із словника person_positions.txt. Такий тип постобробки токенів (використання додаткових словників) називається газетіром та широко застосовується в роботах по NLP.

Для попередньої обробки текстів та покращення подальшої обробки токенів було використано спеціальні газетіри, які було побудовано в рамках дослідження на основі даних з відкритих джерел: список найбільш вживаних прізвищ України [8], найбільш вживані абревіатури української мови для позначення організацій, список станцій метро України, загальні назви комерційних структур, загальні та власні назви вищих навчальних закладів, загальні та власні назви соціальних організацій, загальні назви комерційних організацій, назви роду діяльності людини.

Алгоритм виділення іменованих сутностей

Вперше задачу розпізнавання іменованих сутностей було сформульовано в 1996 році на конференції MUC-6 [6] як завдання знаходження в тексті таких даних, як імена особистостей, назви організацій, час, географічні назви, дати, грошові суми та значення з процентами. Завдання розпізнавання іменованих сутностей проявляється у виявленні та класифікації елементів тексту – слів і послідовностей слів за наведеними вище категоріями.

Наприклад: *[Джек Лондон](PERSON) народився в [Сан Франциско](LOCATION), [Каліфорнція](Location), а не в [Лондоні](LOCATION).*

Різні входження слова Лондон відповідають різним типам іменованих сутностей – географічній назві та прізвищу (власній назві). Вирішення подібних ситуацій робить завдання виділення сутностей нетривіальним для вирішення простим алгоритмічним шляхом.

Дослідниками було запропоновано немало способів виділення іменованих сутностей [7,8,9]. Перші алгоритми, в основному, використовували набір евристик і складених вручну правил [7], які були залежними від мови та стилістики тексту. Більш сучасні підходи використовують алгоритми, засновані на методах машинного навчання з учителем [8,9]. Є навіть нестандартні підходи з використанням генетичних алгоритмів [10] для підбору ознак, за якими буде будуватись модель машинного навчання. Все це дозволяє створювати алгоритми пошуку сутностей без використання експертів у галузі лінгвістики та таких, які можуть не прив'язуватися до конкретної мови.

Було вирішено використати підхід – пошук сутностей за шаблонами. Під час роботи ми створили програмну систему для виокремлення іменованих сутностей з тексту, яка базується на теорії формальних граматик. Такий тип систем є достатньо дієвим у випадку відсутності великої кількості анотованих текстів, що й стало причиною такого непопулярного вибору.

Наш алгоритм виділення сутностей використовує GLR [11] парсер, розроблений у ході роботи. За приклад взято парсер з роботи [12].

Для кожного типу сутності (у цій роботі Персона, Організація та Гео-об’єкт), який потрібно розглядати, ми створюємо набір правил, за якими можна визначити цю сутність. Правила мають наступний вигляд:

```
complex("Person_Full",
        simple(gram("lname"), not(gram("abbr"))),
        simple(gram("fname"), not(gram("abbr")), gnc_match(-1, true)),
        simple(gram("patr"), not(gram("abbr")), gnc_match(-1, true))
    ).
```

Правила задаються декларативно у вигляді Java-об’єктів та компілюються разом з програмою, що додає швидкості виконання.

Тестова версія алгоритму містить 22 правила для виокремлення сутності «персона», 13 правил для «гео-обєкта» та 20 правил для сутності «організація».

Складність алгоритму визначається складністю алгоритму GLR парсера та в найгіршому випадку складає $O(n^3)$.

Оскільки в алгоритмі відсутня недетрмінованість вибору, то час виділення сутностей у тексті буде константним. Середній час виділення сутностей у тексті з ~700 символів становив 50 мілісекунд при 55 правилах різного типу та 5 газетірами (близько 150 слів кожен) для додаткового тегування організацій, прізвищ та персон.

У результаті проекту було створено невеликий корпус анотованих даних на основі новинних статей одного з найбільших інформаційних агентств України – «Західної інформаційної корпорації» (zik.ua) [13] з трьома типами іменованих сутностей: персони, організації та географічні об’єкти.

Основні показники створеного корпусу: кількість анотованих текстів – 122, сутностей типу персона – 1347, сутностей типу організація – 767, сутностей типу географічний об’єкт –617, загальна кількість виділених іменованих сутностей – 2731.

Створення вибірки даних для тестування алгоритму було проведено в застосунку для анотування текстів, який був розроблений спеціально для цієї роботи. Веб-застосунок для зручного анотування даних створено засобами Java, Js та HTML. На рисунку 2 наведено знімок екрану робочого стану застосунку для анотування текстів.

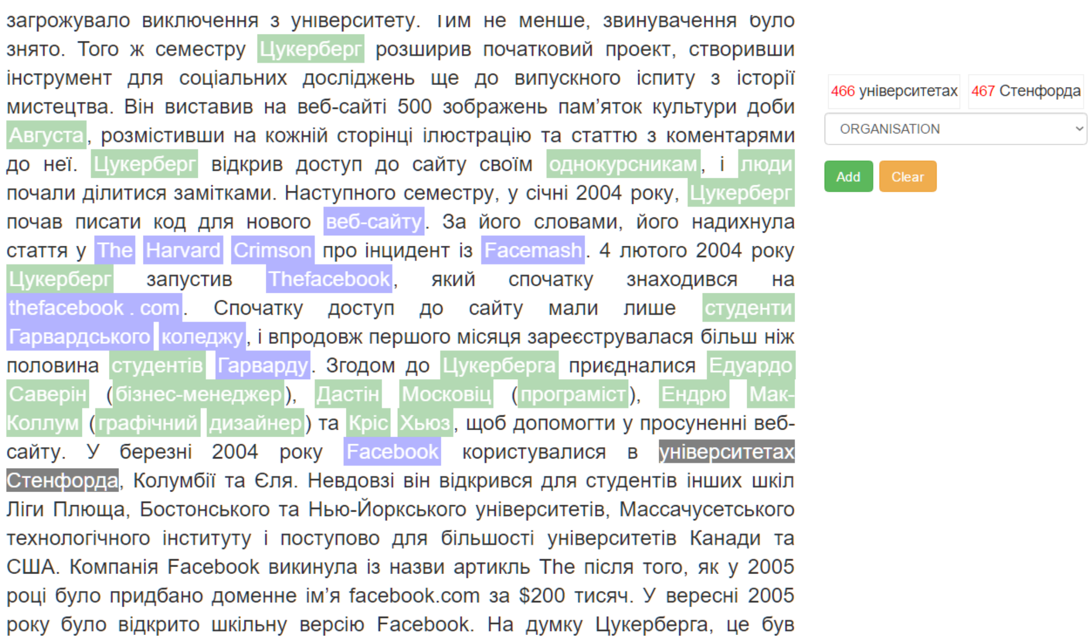


Рис.2. Вигляд екрану робочого стану застосунку для анотування текстів

Програма випадковим чином видає користувачу нерозмічений текст з обраної колекції. Застосунок візуально підсвічує виділені токени, підсвічує уже виділені сутності різними кольорами відповідно до типу сутності. Є можливість видалення проанотованих даних у випадку помилкового проставляння недостовірних даних. Анотовані дані зберігаються у спеціальному форматі в json-файлах. За потреби можна змінити формат збереження анотованих даних у формат збереження BRAT [14], який використовується в наборі стенфордських утиліт.

Оцінку якості роботи алгоритму виділення іменованих сутностей було проведено на корпусі текстів, розміченому в рамках цієї роботи; оцінка алгоритму проводиться в термінах повноти (P, precision), точності (R, recall) та F1-міри (F):

- ***P = (к-ть правильно виділених сутностей)/(кількість всіх виділених)***
- ***R = (к-ть правильно виділених сутностей)/(загальна кількість у колекції)***
- **F=2PR/(P+R) –** ***F1 міра***

Оскільки об’єм тестових даних був невеликий, ми виділили два типи оцінки: строгий і нестрогий.

Строгий – сутність є правильно визначеною, якщо збігаються всі ознаки: межі іменованої сутності – початкова та кінцева позиція та її тип (персона, організація, гео-об’єкт) повністю співпадають з тестовими даними.

Нестрогий – сутність вважається правильно визначеною, якщо хоча б одна з меж іменованої сутності, визначена правильно (початкова або кінцева позиція), та тип співпадають з тестовими даними.

Оцінка результатів побудованого алгоритму не дуже втішна. F-міра для типу сутностей «персона» знаходиться в межах 0.48 при строгому алгоритму оцінки та 0.54 в іншому випадку. Для типу «організація» та «гео» ці показники ще нижчі.

Причиною такого результату є недосконалі правила та недостатня їх кількість, адже при їх створенні ми не користувались допомогою кваліфікованих лінгвістів. Також роботу алгоритму можна покращити, використавши метод обчислення семантичної близькості з роботи [15]. Проте, й такий результат за невеликої кількості правил, не є дуже поганим. Це говорить про те, що алгоритм rule-based NER можна використовувати як допоміжний алгоритм до моделі, заснованій на машинному навчанні або ж як допоміжний алгоритм саме при створенні даних для статистичної моделі.

Висновки

У результаті роботи створено та реалізовано алгоритм пошуку іменованих сутностей в українських текстах.

Важливою складовою є і створені програмні інструменти для зручного анотування та підготовки даних до наступної перевірки або ж використання в навчальних моделях, заснованих на методах машинного навчання. Ці інструменти дозволяють виділяти іменовані сутності та зв'язки між ними в графічному режимі. Було вирішено, для більшої зручності, надавати можливість використовувати дані, отримані за допомогою rule-based алгоритмів, як допоміжні підказки задля збільшення швидкості роботи операторів. Утиліту реалізовано в вигляді веб-застосунку, тому це дає можливість використовувати або інтегрувати його як сервіс для крос платформенної роботи над підготовкою навчальних даних.

Під час роботи над проектом було створено корпус анотованих NER сутностями текстів у розмірі 122 тексти. Проставлені такі види сутностей як *персони*, *організації* та *географічні об’єкти*. Корпус складається з 2731 іменованої сутності.

Література

1. «CoNLL 2017 | CoNLL.». [Електронний ресурс].URL: http://www.conll.org/ (Дата звернення 4 червня. 2017).
2. «Message Understanding Conference - 6: A Brief History - NYU.» [Електронний ресурс]. URL: http://nlp.cs.nyu.edu/muc/muc6-history-coling.ps. (Дата звернення 27травня. 2017).
3. David Crystal. «A dictionary of linguistics and phonetics (sixth edition)», 2008.
4. «PCRE - Perl Compatible Regular Expressions.» [Електронний ресурс] URL: http://www.pcre.org/. (Дата звернення 04.06.2017)
5. «LanguageTool.Org.» [Електронний ресурс] URL: https://www.languagetool.org/. (Дата звернення 04.06.2017)
6. MUC-6. [Електронний ресурс].URL: http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html (Дата звернення 29.05.2017)
7. A Borthwick- Ph. D. Thesis New York University, 1999 - A Maximum Entropy Approach to Named Entity Recognition
8. David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. EMNLP.
9. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), Vol. 4. Association for Computational Linguistics, Stroudsburg, PA, USA, 168-171.
10. Huong Thanh Le, Luan Van Tran, Xuan Hoai Nguyen, and Thi Hien Nguyen. 2015. Optimizing Genetic Algorithm in Feature Selection for Named Entity Recognition. In Proceedings of the Sixth International Symposium on Information and Communication Technology (SoICT 2015). ACM, New York, NY, USA, 11-16.
11. «GLR parser - Wikipedia.» [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/GLR_parser. (Дата звернення 29.05.2017)
12. "Yargy is a GLR-parser, that uses russian morphology for facts extraction process, and written in pure python". [Електронний ресурс]. URL: https://github.com/bureaucratic-labs/yargy (Дата звернення 29.05.2017)
13. «zik.ua Analytics - Market Share Stats & Traffic Ranking - SimilarWeb.» [Електронний ресурс]. URL: https://www.similarweb.com/website/zik.ua. (Дата звернення 29.05.2017)
14. «brat rapid annotation tool.» [Електронний ресурс]. URL: http://brat.nlplab.org/. (Дата звернення 29.05.2017)
15. Метод обчислення семантичної близькості для слів природної мови / А. В. Анісімов, М. М. Глибовець, О. О. Марченко, В. К. Кисенко // Наукові записки НаУКМА. Комп'ютерні науки. - 2011. - Т. 125. - С. 8-12.

Literatura

1. «CoNLL 2017 | CoNLL.». [Elektr. Resurs].URL: http://www.conll.org/
2. «Message Understanding Conference - 6: A Brief History - NYU.» [Elektr. Resurs]. URL: http://nlp.cs.nyu.edu/muc/muc6-history-coling.ps
3. David Crystal. "A dictionary of linguistics and phonetics (sixth edition)", 2008.
4. «PCRE - Perl Compatible Regular Expressions.» [Elektr. Resurs] URL: http://www.pcre.org/
5. «LanguageTool.Org.» [Elektr. Resurs] URL: https://www.languagetool.org/
6. MUC-6. [Електронний ресурс].URL: http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
7. A Borthwick- Ph. D. Thesis New York University, 1999 - A Maximum Entropy Approach to Named Entity Recognition
8. David Pierce and Claire Cardie. 2001. Limitations of co-training for natural language learning from large datasets. EMNLP.
9. Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 (CONLL '03), Vol. 4. Association for Computational Linguistics, Stroudsburg, PA, USA, 168-171.
10. Huong Thanh Le, Luan Van Tran, Xuan Hoai Nguyen, and Thi Hien Nguyen. 2015. Optimizing Genetic Algorithm in Feature Selection for Named Entity Recognition. In Proceedings of the Sixth International Symposium on Information and Communication Technology (SoICT 2015). ACM, New York, NY, USA, 11-16.
11. «GLR parser - Wikipedia.» [Elektr. Resurs]. URL: https://en.wikipedia.org/wiki/GLR_parser
12. "Yargy is a GLR-parser, that uses russian morphology for facts extraction process, and written in pure python". [Elektr. Resurs]. URL: https://github.com/bureaucratic-labs/yargy
13. «zik.ua Analytics - Market Share Stats & Traffic Ranking - SimilarWeb.» [Електронний ресурс]. URL: https://www.similarweb.com/website/zik.ua
14. «brat rapid annotation tool.» [Електронний ресурс]. URL: http://brat.nlplab.org/
15. Metod obchislenny semantichnoi blizkosti dly sliv prirodnoy movi / A. V. Anisimov, M. M. Glybovets, O. O. Marchenko, B. K. Kislenko // Scientific notes of NaUKMA. Computer Science. - 2011. - T. 125. - C. 8-12.

RESUME

A.M. Glybovets

Automated search of named entities in unmarked ukrainian texts

As a result of the work, the algorithm of the search for named entities in Ukrainian texts was created and implemented.

An important component is the created software tools for easy annotation and preparation of data for the next check or use in automatic learning models based on methods of machine learning. These tools allow you to allocate named entities and links between them in graphical mode. It was developed for greater convenience to provide the opportunity to use the data obtained using rule-based algorithms as auxiliary hints to increase the speed of the operators. The utility is implemented as a web application, so it enables you to use or integrate it as a service for cross platform work on the preparation of training data.

The evaluation of the results of the algorithm is not very comforting. The F-measure for a person's entity type is within 0.48 with a strict evaluation algorithm and 0.54 otherwise; for the "organization" and "geo" type, these figures are even lower.

The reason for this result is imperfect rules, but their number is insufficient, because at the time of their creation we did not use the help of qualified linguists. Also, the algorithm's work can be improved by using the method of calculating semantic proximity of the words. However, this result for a small number of rules is not very bad. This suggests that the rule-based NER algorithm can be used as an auxiliary algorithm for a model based on machine learning or as an auxiliary algorithm precisely when creating data for a statistical model.

During the work on the project, a body of annotated NERs was created for the content of texts in the amount of 122 texts. There are such kinds of entities as persons, organizations and geographical objects. The body consists of 2,731 named entities.

Надійшла до редакції 19.09.2017